```
//
//  Account.h
//  ATM
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <Foundation/Foundation.h>

@interface Account : NSObject {
}

@property (assign, nonatomic) unsigned long long balance;

@end
```

```
//
//  Account.m
//  ATM
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import "Account.h"

@implementation Account

@synthesize balance=_balance;

- (id)init
{
    if ((self = [super init])) {
        self.balance = 0;
    }
    return self;
}

@end
```

```
//
//  Controller.h
//  ATM
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <Foundation/Foundation.h>
#import "Account.h"

@interface Controller : NSObject {
}

@property (nonatomic, retain) Account *account;
@property (assign, nonatomic) unsigned long long amount;
@property (nonatomic, retain) IBOutlet UILabel *balanceLabel;
@property (nonatomic, retain) IBOutlet UILabel *depositLabel;

- (IBAction)clear:(id)sender;
- (IBAction)deposit:(id)sender;
- (IBAction)digit:(id)sender;
- (void)show;

@end
```

```
//
//  Controller.m
//  ATM
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import "Controller.h"

@implementation Controller

@synthesize account=_account;
@synthesize amount=_amount;
@synthesize balanceLabel=_balanceLabel;
@synthesize depositLabel=_depositLabel;

- (void)awakeFromNib
{
    // create account
    Account *account = [[Account alloc] init];
    self.account = account;
    [account release];

    // show UI
    [self show];
}

- (IBAction)clear:(id)sender
{
    // clear input
    self.amount = 0;
    [self show];
}

- (IBAction)deposit:(id)sender
{
    // deposit amount
    self.account.balance += self.amount;

    // clear input
    self.amount = 0;
    [self show];
}

```

```
49. - (IBAction)digit:(id)sender
50. {
51.     // append digit to amount
52.     UIButton *b = (UIButton *)sender;
53.     self.amount = self.amount * 10 + b.tag;
54.     [self show];
55. }
56.
57. - (void)show
58. {
59.     // show balance
60.     self.balanceLabel.text = [NSString stringWithFormat:@"$%llu", self.account.balance];
61.
62.     // show input
63.     self.depositLabel.text = [NSString stringWithFormat:@"$%llu", self.amount];
64. }
65.
66. - (void)dealloc
67. {
68.     [_account release];
69.     [_balanceLabel release];
70.     [_depositLabel release];
71.     [super dealloc];
72. }
73.
74. @end
```

```
//
//  GesturesViewController.h
//  Gestures
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates gestures.
//

#import <UIKit/UIKit.h>

@interface GesturesViewController : UIViewController <UIAlertViewDelegate> {
}

@property (assign, nonatomic) BOOL alertInProgress;
@property (nonatomic, retain) IBOutlet UIImageView *imageView;
@property (assign, nonatomic) int index;
@property (nonatomic, retain) NSArray *tommies;

- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex;
- (void)handleLongPress:(UILongPressGestureRecognizer *)sender;
- (void)handleSwipe:(UISwipeGestureRecognizer *)sender;


@end
```

```m
//
//  GesturesViewController.m
//  Gestures
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates gestures.
//

#import "GesturesViewController.h"

@implementation GesturesViewController

@synthesize alertInProgress=_alertInProgress;
@synthesize imageView=_imageView;
@synthesize index=_index;
@synthesize tommies=_tommies;

- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex
{
    self.alertInProgress = NO;
}

- (void)handleLongPress:(UILongPressGestureRecognizer *)sender
{
    if (!self.alertInProgress) {
        self.alertInProgress = YES;
        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Hey!"
                                                            message:@"Stop that."
                                                           delegate:self
                                                  cancelButtonTitle:@"Fine"
                                                  otherButtonTitles:nil];
        [alertView show];
        [alertView release];
    }
}

- (void)handleSwipe:(UISwipeGestureRecognizer *)sender
{
    // handle swipe
    UISwipeGestureRecognizerDirection direction = [(UISwipeGestureRecognizer *)sender direction];
    switch (direction) {

            // ignore up, down
        case UISwipeGestureRecognizerDirectionUp:
        case UISwipeGestureRecognizerDirectionDown:
```

```
49.             break;
50.
51.             // left
52.         case UISwipeGestureRecognizerDirectionLeft:
53.             self.index = (self.index + 1) % [self.tommies count];
54.             break;
55.
56.             // right
57.         case UISwipeGestureRecognizerDirectionRight:
58.             self.index = (self.index + [self.tommies count] - 1) % [self.tommies count];
59.             break;
60.     }
61.
62.     // update Tommy
63.     self.imageView.image = [UIImage imageNamed:[self.tommies objectAtIndex:self.index]];
64. }
65.
66. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
67. {
68.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
69. }
70.
71. - (void)viewDidLoad
72. {
73.     [super viewDidLoad];
74.
75.     // keep track of alerts
76.     self.alertInProgress = NO;
77.
78.     // prepare Tommies
79.     self.tommies = [[NSArray alloc] initWithObjects:@"tommy1.jpg", @"tommy2.jpg", @"tommy3.jpg", nil];
80.     self.index = 0;
81.
82.     // listen for long press
83.     UILongPressGestureRecognizer *longpressGesture = [[UILongPressGestureRecognizer alloc] initWithTarget:self action:@selector(
    handleLongPress:)];
84.     [self.imageView addGestureRecognizer:longpressGesture];
85.     [longpressGesture release];
86.
87.     // listen for right swipe
88.     UISwipeGestureRecognizer *swipeGesture = [[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector(handleSwipe:)];
89.     swipeGesture.direction = UISwipeGestureRecognizerDirectionRight;
90.     [self.imageView addGestureRecognizer:swipeGesture];
91.     [swipeGesture release];
92.
93.     // listen for left swipe
94.     swipeGesture = [[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector(handleSwipe:)];
95.     swipeGesture.direction = UISwipeGestureRecognizerDirectionLeft;
```

```
 96.     [self.imageView addGestureRecognizer:swipeGesture];
 97.     [swipeGesture release];
 98. }
 99.
100. - (void)dealloc
101. {
102.     [_imageView release];
103.     [_tommies release];
104.     [super dealloc];
105. }
106.
107. @end
```

```
//
//  RootViewController.h
//  Plist
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates property lists.
//

#import <UIKit/UIKit.h>

@interface RootViewController : UITableViewController {
}

@property (nonatomic, retain) NSArray *words;

@end
```

```m
//
//  RootViewController.m
//  Plist
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates property lists.
//

#import "RootViewController.h"

@implementation RootViewController

@synthesize words=_words;

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // allocate cell, reusing if possible
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] autorelease];
    }

    // configure cell
    cell.selectionStyle = UITableViewCellSelectionStyleNone;
    cell.textLabel.text = [self.words objectAtIndex:[indexPath row]];

    return cell;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [self.words count];
}

```

```
49. - (void)viewDidLoad
50. {
51.     [super viewDidLoad];
52. 
53.     // load words
54.     NSString *path = [[NSBundle mainBundle] pathForResource:@"words" ofType:@"plist"];
55.     NSArray *words = [[NSArray alloc] initWithContentsOfFile:path];
56.     self.words = words;
57.     [words release];
58. }
59. 
60. @end
```

```
//
//  RootViewController.h
//  Sqlite
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates SQLite.
//

#import <UIKit/UIKit.h>
#import "sqlite3.h"

@interface RootViewController : UITableViewController {
}

@property (nonatomic, retain) NSArray *words;

@end
```

```
//
//  RootViewController.m
//  Sqlite
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates SQLite.
//

#import "RootViewController.h"

@implementation RootViewController

@synthesize words=_words;

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // allocate cell, reusing if possible
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] autorelease];
    }

    // configure cell
    cell.selectionStyle = UITableViewCellSelectionStyleNone;
    cell.textLabel.text = [self.words objectAtIndex:[indexPath row]];

    return cell;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [self.words count];
}

```

```
49. - (void)viewDidLoad
50. {
51.     [super viewDidLoad];
52. 
53.     // prepare for words
54.     NSMutableArray *words = [[NSMutableArray alloc] init];
55. 
56.     // connect to SQLite database
57.     sqlite3 *db;
58.     NSString *path = [[NSBundle mainBundle] pathForResource:@"words" ofType:@"sqlite"];
59.     sqlite3_open([path UTF8String], &db);
60. 
61.     // select four-letter words words
62.     NSString *sql = @"SELECT word FROM words WHERE LENGTH(word) = 4";
63.     sqlite3_stmt *statement;
64.     sqlite3_prepare_v2(db, [sql UTF8String], -1, &statement, nil);
65. 
66.     // iterate over results
67.     while (sqlite3_step(statement) == SQLITE_ROW) {
68.         char *c = (char *) sqlite3_column_text(statement, 0);
69.         NSString *s = [[NSString alloc] initWithUTF8String:c];
70.         [words addObject:s];
71.         [s release];
72.     }
73. 
74.     // close database
75.     sqlite3_close(db);
76. 
77.     // retain words
78.     self.words = words;
79.     [words release];
80. }
81. 
82. @end
```

```
//
//  FirstViewController.h
//  TabBar
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Tab Bar Application template.
//

#import <UIKit/UIKit.h>


@interface FirstViewController : UIViewController {

}

@end
```

```m
//
//  FirstViewController.m
//  TabBar
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Tab Bar Application template.
//

#import "FirstViewController.h"


@implementation FirstViewController

/*
// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
}
*/

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}


- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in use.
}


- (void)viewDidUnload
{
    [super viewDidUnload];

    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

```

```
49.
50. - (void)dealloc
51. {
52.     [super dealloc];
53. }
54.
55. @end
```

```
1. //
2. //  SecondViewController.h
3. //  TabBar
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Tab Bar Application template.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14.
15. @interface SecondViewController : UIViewController {
16.
17. }
18.
19. @end
```

```m
//
//  SecondViewController.m
//  TabBar
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Tab Bar Application template.
//

#import "SecondViewController.h"


@implementation SecondViewController

/*
// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
}
*/

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}


- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in use.
}


- (void)viewDidUnload
{
    [super viewDidUnload];

    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

```

```
49.
50.  - (void)dealloc
51.  {
52.      [super dealloc];
53.  }
54.
55.  @end
```

```
//
//  TransformationsViewController.h
//  Transformations
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates affine transformations of Tommy.
//

#import <UIKit/UIKit.h>

@interface TransformationsViewController : UIViewController {
    CGPoint _translation;
}

@property (nonatomic, retain) IBOutlet UIImageView *imageView;
@property (assign, nonatomic) float scale;

- (void)handlePan:(UIPanGestureRecognizer *)sender;
- (void)handlePinch:(UIPinchGestureRecognizer *)sender;

@end
```

```m
//
//  TransformationsViewController.m
//  Transformations
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates affine transformations of Tommy.
//

#import "TransformationsViewController.h"

@implementation TransformationsViewController

@synthesize imageView=_imageView;
@synthesize scale=_scale;

- (void)handlePan:(UIPanGestureRecognizer *)sender
{
    // translate Tommy
    CGPoint translation = [sender translationInView:self.imageView];
    CGAffineTransform scale = CGAffineTransformMakeScale(self.scale, self.scale);
    CGAffineTransform translate = CGAffineTransformMakeTranslation(_translation.x + translation.x * self.scale, _translation.y + translation.y * self.scale);
    sender.view.transform = CGAffineTransformConcat(scale, translate);

    // remember translation once done panning
    if (sender.state == UIGestureRecognizerStateEnded) {
        _translation.x += translation.x * self.scale;
        _translation.y += translation.y * self.scale;
    }
}

- (void)handlePinch:(UIPinchGestureRecognizer *)sender
{
    // scale Tommy
    CGFloat factor = [sender scale];
    CGAffineTransform scale = CGAffineTransformMakeScale(self.scale * factor, self.scale * factor);
    CGAffineTransform translate = CGAffineTransformMakeTranslation(_translation.x, _translation.y);
    sender.view.transform = CGAffineTransformConcat(scale, translate);

    // remember scale once done pinching
    if (sender.state == UIGestureRecognizerStateEnded) {
        self.scale *= factor;
    }
}

```

```
48. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
49. {
50.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
51. }
52. 
53. - (void)viewDidLoad
54. {
55.     [super viewDidLoad];
56. 
57.     // initial scale
58.     self.scale = 1.0;
59. 
60.     // initial translation
61.     _translation.x = 0.0;
62.     _translation.y = 0.0;
63. 
64.     // listen for pan
65.     UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc] initWithTarget:self action:@selector(handlePan:)];
66.     [self.imageView addGestureRecognizer:panGesture];
67.     [panGesture release];
68. 
69.     // listen for pinch
70.     UIPinchGestureRecognizer *pinchGesture = [[UIPinchGestureRecognizer alloc] initWithTarget:self action:@selector(handlePinch:)];
71.     [self.imageView addGestureRecognizer:pinchGesture];
72.     [pinchGesture release];
73. }
74. 
75. - (void)dealloc
76. {
77.     [_imageView release];
78.     [super dealloc];
79. }
80. 
81. @end
```